بوابة الإنترنت الرقمية

دورة تعلم برمجة تطبيقات الأندرويد

-المحاضرة السابعة-

Intents

هذه الدورة برعاية

متجر البلوتوث

www.albluetooth.com

كل الشكر والتقدير للدكتور المهندس

عبدالحليم دندوش في توجيهي لنشر الفائدة للجميع ..

في هذه المحاضرة - بإذن الله - سوف نقوم بشرح مفهوم عميق وأساسي جدا في مجال برمجة تطبيقات الأندرويد ، سبق وأن تم التقديم لمفهوم ال Intents في محاضرات سابقة وهي التي سوف تأخذ جل وقت محاضرتنا هذه ، كما سوف نأخذ أيضا بعض الميزات مثل نافذة التنويه alertDialog ..

الـIntents هي بإختصار :حدث يقوم بنقل التحكم من Activity لأخرى .. "يستخدم للإستدعاء"

إستخداماته :

للـ intents إستخدامات عديدة ، أهمها والتي قد تستخدمها في برمجياتك :

1 - الإنتقال من activity إلى activity أخرى "من واجهة لأخرى" وهذا النوع لا يخلو أي برنامج منه

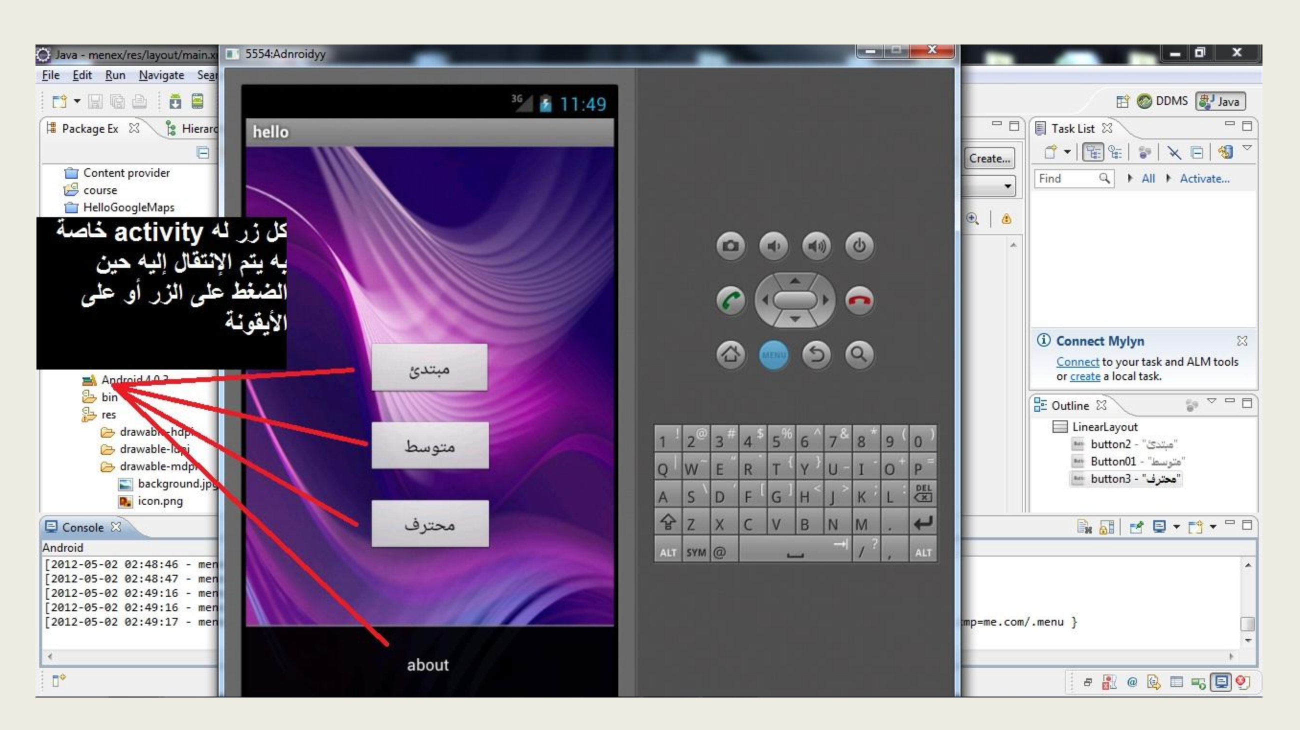
2 - إستدعاء برنامج خارجي أو خدمة من خدمات أندرويد

سوف نقوم بهذه المحاضرة بدراسة الحالة الأولى بشكل موسّع ..

لنأخذ سيناريو على كل إستخدام :

لنفرض أنه لدينا لعبة تعليمية ما ، هذه اللعبة عبارة عن عمليات رياضية بسيطة "جمع وطرح وهكذا" ، لدينا واجهة بداية لهذه اللعبة ، هذه الواجهة تحتوي على الصعوبة التي نريد إختيارها "مبدئ ، متوسط ، محترف " ومن زر ال menu يوجد أيقونة about ..

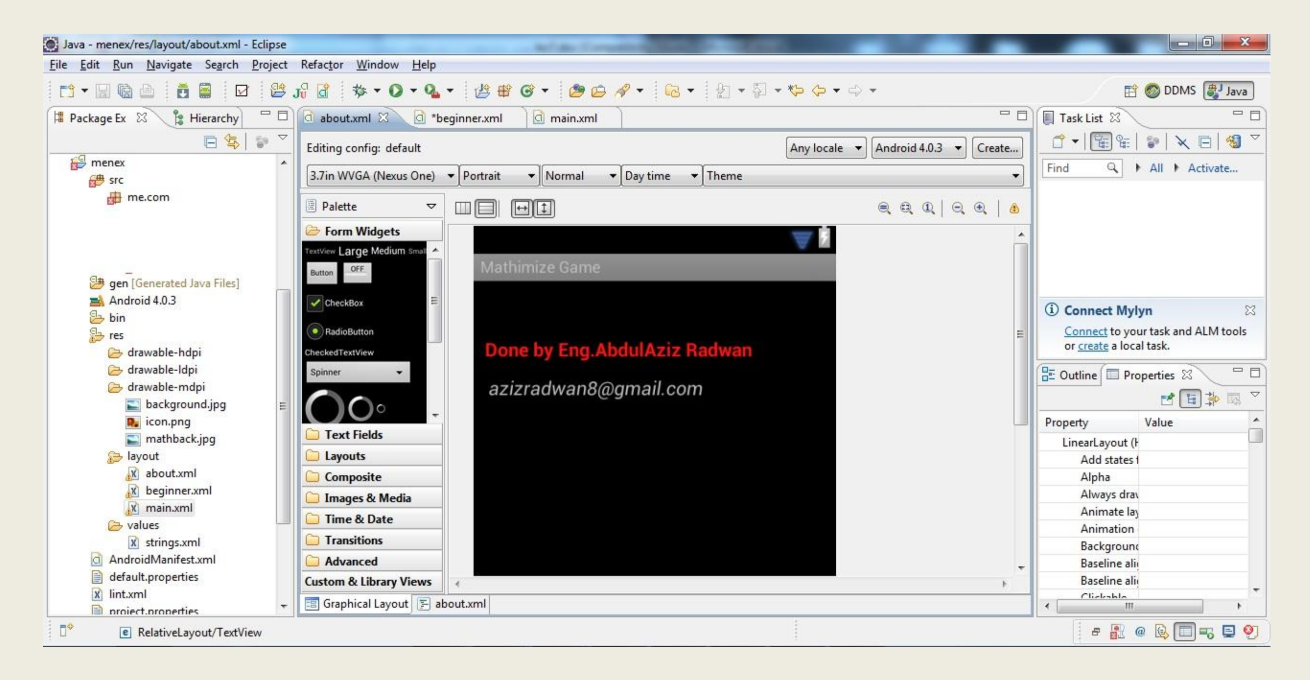
بالتالي مبدئيا وبالتخطيط للمشروع وقبل البدء به نلاحظ أننا سنحتاج إلى activity رئيسية و3 activities لكل صعوبة واحدة ، وواجهة أخيرة من أجل الـabout ..



نقوم بتصميم كل activity على حدة ، كما في المحاضرات السابقة .. فنقوم هنا مثلا بعمل واجهة تحتوي البيانات الخاصة بـabout وهي إسم المبرمج وإيميله وموقعه وشعاره – على سبيل المثال – ونقوم بإنشاء واجهات أخرى من أجل المبتدئ والمتوسط والمحترف .. "سوف أقوم في هذه المحاضرة بإنشاء المبتدئ فقط" ..

نقوم بالبداية بتصميم الوجهات كما أخذنا في المحاضرات السابقة تماما "لا شيء جديد"

واجهة الـabout :



واجهة المبتدئ "Beginner" :



ملاحظة : من أجل سهولة ترتيب العناصر فوق بعضها ينصح بإستخدام الـRelative Layout وسبق شرحها في المحاضرات الثانية والثالثة

نلاحظ في هذه الواجهة توجد عمليات حسابية وبها فراغات ، على اللاعب أن يقوم بتعبئة الفراغات تعبئة صحيحة ومن ثم الضغط على result .. وهنا لدينا حالتين :

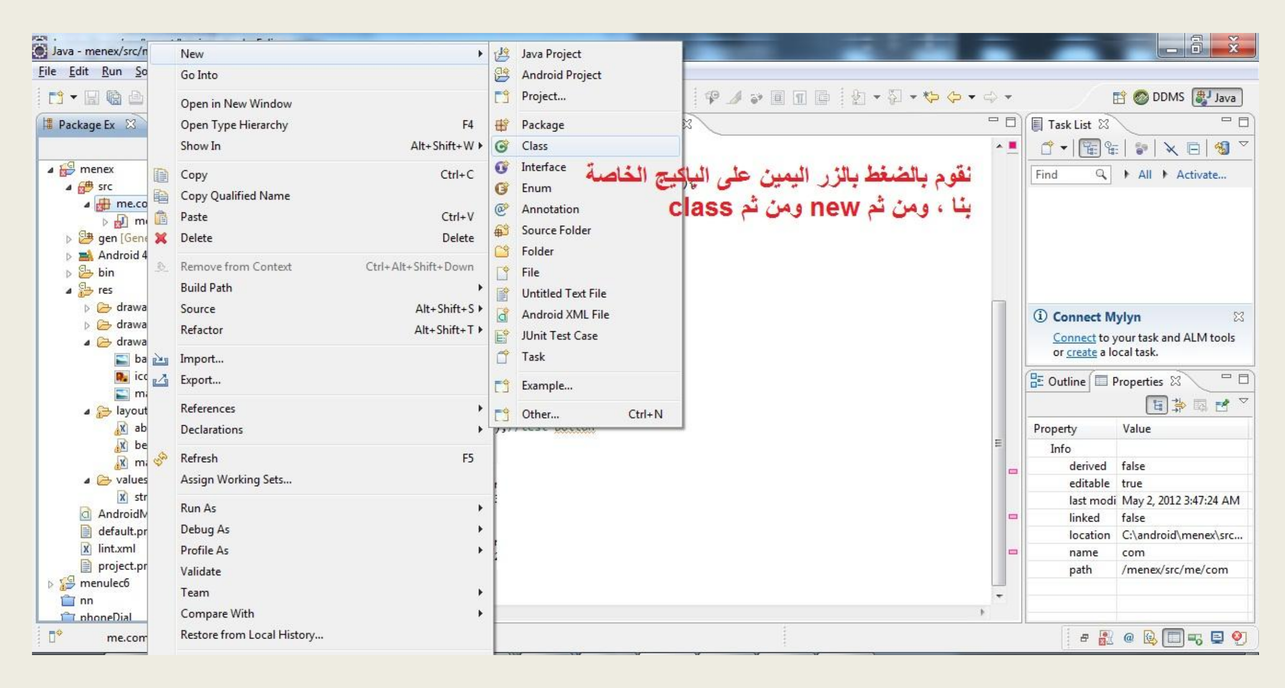
الحالة الأولى : أن تكون جميع الأجوبة صحيحة : في هذه الحالة يتم إظهار نافذة صغيرة تحوي عدد المحاولات ويتم إعادة اللعبة مرة أخرى .

الحالة الثانية : هناك عدد من الأجوبة صحيحة وعدد من الأجوبة خاطئة : في هذه الحالة يتم تلوين الأجوبة الصحيحة باللون الأزرق ويتم تلوين الأجوبة الخاطئة باللون الأحمر ويتم إحتساب عدد محاولات خاطئة = 1 .

"هذا العمل من إختصاص المبرمج وهو نفسه سواء بالجافا او بالسي بلس بلس أو حتى بالباسكال ، وليس من إختصاص الأندرويد بحد ذاته .. لذلك سوف أقوم بالعمل به بشكل سريع حيث أنه من غير إختصاص الدورة وقد قمت بالأعلى بشرح الخوارزمية وما تبقى هو برمجة بحته "

نقوم بالطبع بكتابة الخوارزمية الخاصة بهذا الحل في ملف الjava. ، ومن هنا علينا التنويه أننا نحتاج إلى ملفات java. بعدد الواجهات التي لدينا ، بحيث كل activity لكي تعمل بشكل سليم يجب علينا أن نوفر لها جزء تصميمي وهو ملف ال xml.
وملف البرمجة وهو ال java. سواء كانت تحتاج إلى برمجة مثل اللعبة أو كان لا يحتاج مثل واجهة ال about .. سوف نتطرق إلى الخطوات في آخر المحاضرة

نقوم الآن بإنشاء ملفات الـjava. الخاصة بكل واجهة قمنا بتصميمها :



نقوم بكتابة إسم الـclass المطلوب إنشاؤه ، يفضل كتابة أسماء ذات علاقة بالعمل المطلوب ، مثلا للتصميم about.xml نقوم بعمل ملف برمجة بإسم about.java

سوف يتم فتح كلاس فارغ إلا من إسمه

```
package me.com;
public class about {
}
```

، نقوم بهذا الclass بكتابة الكود البرمجي الخاص بهذه الواجهة ، مثلا في حالة الـabout ليس هناك أي عمل برمجي ، لذلك نجعل الصفحة تقوم بإظهار التصميم فقط ، وذلك بخطوات ثابتة وهي :

1 - جعل الـclass يرث كلاس الـactivity

2 - كتابة الطريقة onCreate وبها يقوم بإظهار التصميم المطلوب فور إستدعاء هذا الملف ..

فيصبح لدينا الملف بالشكل التالي :

```
package me.com;

import android.app.Activity;
import android.os.Bundle;

public class about extends Activity {


       public void onCreate(Bundle savedInstanceState) {
              super.onCreate(savedInstanceState);
              setContentView(R.layout.about);
          }

}
```

نلاحظ أن إسم ملف التصميم الخاص بي هو about وقد تم وضعه في الطريقة بهذا الشكل ، لا يوجد بالملف البرمجي أي تعليمات أخرى لذلك لن ينفذ أي شيء بعد إستدعاء التصميم ..

نقوم بإنشاء class آخر من أجل برمجة خوارزمية اللعبة على المستوى البسيط وربطه بالواجهة الخاصة بالمستوى البسيط وإظهارها ، فيكون الكود على الشكل التالي :

```
public class beginner extends Activity {
      int error = 0;


       public void onCreate(Bundle savedInstanceState) {

             super.onCreate(savedInstanceState);
              setContentView(R.layout.beginner);


              final EditText Et1 = (EditText) findViewById(R.id.ans5);
              final EditText Et2 = (EditText) findViewById(R.id.ans25);
              Button check = (Button) findViewById(R.id.check);
```

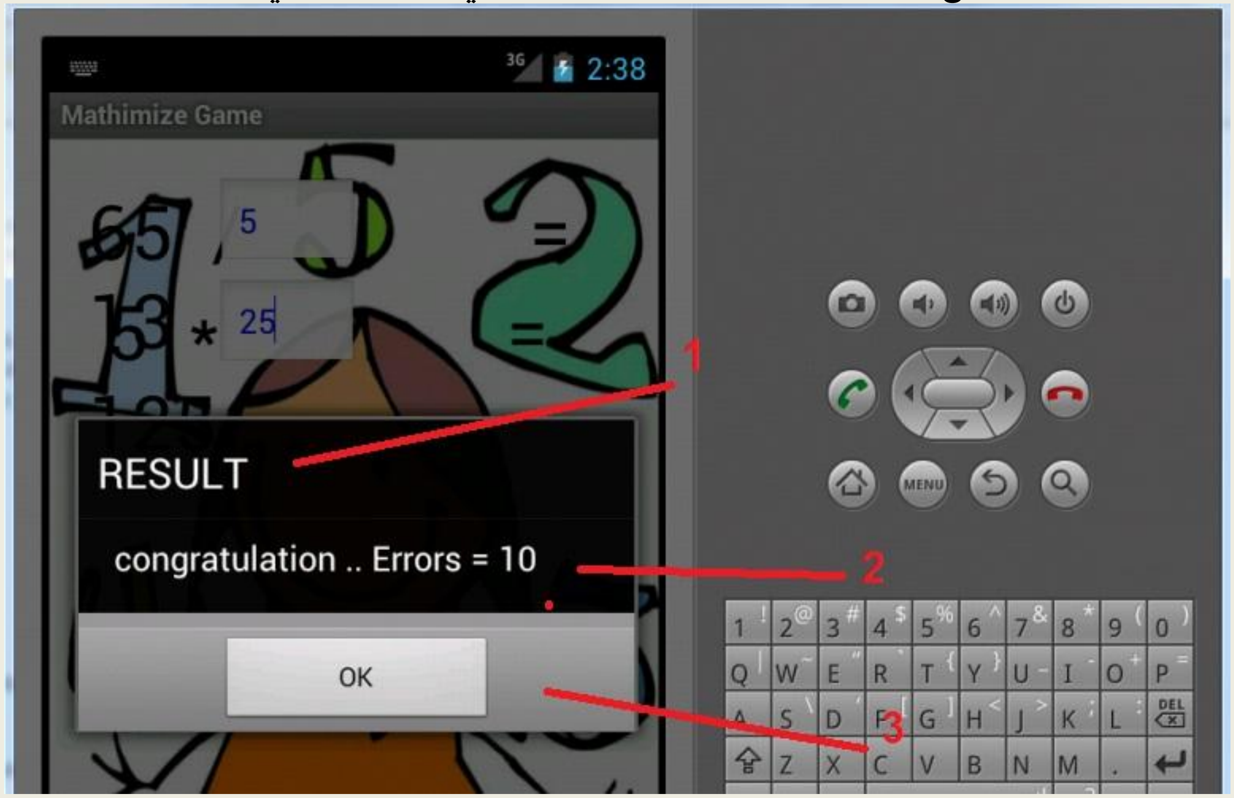
قمنا بجعل الملف - ككل الملفات التي تكون مرتبطة مع تصميم معيّن- يقوم بإظهار التصميم عند أول إستدعاء له –كما ذكرنا في الخطوات السابقة ، كما قمنا بتعريف الخانات التي نريد أن نقوم بإختبارها وهي الفراغات التي سوف يقوم

اللاعب بملأها والزر الذي سوف يعيد النتيجة ، قمنا أيضا بتعريف عداد للأخطاء التي قد يخطئ بها اللاعب..

لاحظ أننا وضعنا مجال عداد الأخطاء error على الـclass ككل لإننا نريده شامل لجميع الأخطاء وليس على مجال الزر فقط ، أي ان الأخطاء تحسب ليس على كل ضغطة زر ، بل على اللعبة بشكل كامل "لمزيد من المعلومات راجع المجالات في java "

نقوم الآن بتعريف آلية عمل الزر ، بحيث يقارن المدخلات من اللاعب مع الأجوبة الصحيحة ، يقوم بتغيير اللون ويقوم بحساب الأخطاء ، يقوم بإظهار نافذة توضح الأخطاء حين إنتهاء اللعبة ..
جميع ما سبق قد تم إعطاؤه سابقا ما عدا إظهار النافذة ، لذا سنقوم بأخذ الخطوات قبل وضع الكود ..

نافذة التنويه أو الAlertDialog نراها واضحة في الشكل التالي :



نستطيع إستدعاء هذه النافذة متى ما شئنا ، عند الضغط على زر معيّن أو على أيقونة معيّنة ، عند مرور وقت معيّن ، عند الخروج من البرنامج أو عند الدخول إليه ، عند حدوث حدث معيّن – كما في مشروعنا عند الإجابة على جميع الأجوبة بشكل صحيح ..

نستطيع إنشاء النافذة بإستخدام التعليمة :

```
AlertDialog alertDialog = new AlertDialog.Builder(Main.this)

 .create();
```

حيث Main هو إسم الClass الذي نرغب بأن تظهر النافذة به

ولها – كما نرى في الشكل أعلاه – الخواص المرقمة :

1 -وهو عنوان النافذة ويمكن وضعه بالتعليمة التالية:

```
alertDialog.setTitle("RESULT ");
```

2 -محتوى النافذة : وهي رسالة نستطيع وضعها هنا مثل : شكرا لزيارتك أو مرحبا بك أو عبارات تنويه أو النتيجة مثل حالتنا هنا في البرنامج ويمكن وضعها بالتعليمة التالية :

```
alertDialog.setMessage("congratulation ");
```

3 - الزر ، يوضع في آخر النافذة ويمكن برمجته مثل أي زر آخر ويمكن الكتابة عليه ويمكن وضعه بالتعليمات التالية :

```
alertDialog.setButton("OK", new DialogInterface.OnClickListener()
{
    public void onClick(DialogInterface dialog, int which) {
      //here you can add functions
  } });
```

حيث OK هي الكلمة التي سوف تظهر على الزر ، والمكان المؤشر بالأحمر هو المكان الذي سوف نقوم بكتابة الكود الخاص بهذا الزر فيه ..

وأخيرا بعد الإنتهاء من جميع ما سبق نقوم بكتابة تعليمة إظهار النافذة وهي :

```
alertDialog.show();
```

فتظهر لنا النافذة كما في الشكل السابق ..

الآن نعود إلى الكود الخاص بهذا البرنامج بعد أن أصبحت الرؤية كاملة ونضع الكود الخاص بالزر :

```
    check.setOnClickListener(new OnClickListener() {
                    /**UsetheonClicktocalltheexistingIntentcode*/
                    public void onClick(View v){
                          boolean fin = true;
if(Et1.getText().toString().equals("5")){
Et1.setTextColor(Color.BLUE);}

else {
      fin=false;
      error ++;
      Et1.setTextColor(Color.RED);


}
if(Et2.getText().toString().equals("25")){
Et2.setTextColor(Color.BLUE);}
else {
      fin=false;

      error ++;
      Et2.setTextColor(Color.RED);
}

if(fin==true){
    AlertDialog alertDialog = new AlertDialog.Builder(beginner.this).create();
    alertDialog.setTitle("RESULT");
    alertDialog.setMessage("congratulation .. Errors = " + error);
    alertDialog.setButton("OK", new DialogInterface.OnClickListener() {
        public void onClick(DialogInterface dialog, int which) {
   error = 0;
   Et1.setText("?", EditText.BufferType.EDITABLE );
   Et2.setText("?", EditText.BufferType.EDITABLE );

      } });
    alertDialog.show();
```

الكود جدا بسيط ، يقوم بالمقارنة بحيث إذا كان الناتج مثل المطلوب يقوم بتلوينه بالأزرق ، عدا ذلك يقوم بتعطيل الفوز باللعبة عن طريق وضع المتحول fin = false ويقوم بزيادة عداد الأخطاء 1+ ، ويقوم بتغيير اللون إلى الأحمر ..

في حال حل جميع الأجوبة بشكل صحيح يتم الدخول إلى تعليمة if الأخيرة وبالتالي يقوم بإظهار النافذة الخاصة بالنتيجة ومن ثم تصفير عداد الأخطاء ومسح الفراغات من أجل إعادة اللعب مرة أخرى ..

الآن نعود إلى صلب الموضوع الخاص بنا .. بعد هذا كله ، كيف نقوم بربط الواجهات مع بعضها البعض ؟

الجواب : بإستخدام ال intent وذلك بإستخدام الخطوات التالية :

1 - تأكد من أنك قمت بربط كل تصميم بالملف البرمجي الخاص به "يجب أن يكون لكل تصميم ملف برمجي"

2 - توجه إلى الملفات التي تريد ربطها مع بعضها "مثلا في حالة مشروعنا الخاص باللعبة لدينا الواجهة الرئيسية وهي التي نريد ربطها مع الملفات الأخرى :نريد ربط الواجهة الرئيسية مع ملف المبتدئ ومع المتوسط ومع المحترف ومع الabout ..

3 - نقوم بكتابة الكود التالي لإستدعاء الواجهة المطلوبة :

```
Intent test=new Intent(this,example.class);

startActivity(test);
```

بحيث يكون إسم الملف البرمجي المراد إستدعاؤه مكان الـ example

بالتالي من أجل إعادة ربط الأفكار :

نقوم بتصميم الواجهات ونقوم بربط كل واجهة بالملف البرمجي الخاص بها ، حين نرغب بإستدعاء واجهة ما نقوم بإنشاء intent ينقل التحكم إلى ذلك الملف البرمجي الذي بدوره يقوم بإظهار التصميم على الجهاز الخاص بك ..

أتمنى أن تكون الصورة قد وضحت

من المفترض الآن أن يكون البرنامج قد إكتمل وأصبح جاهزا للعمل ، إلا أننا نتفاجئ حين تشغيل البرنامج بعدم إستطاعتنا الوصول إلى الواجهات التي قمنا بالتوجيه إليها وتظهر لنا رسالة بأن البرنامج قد صادف خطأ ويجب إغلاقه ! ، أين المشكلة ؟

تكمن المشكلة بأنه أي ملف برمجي نقوم بإنشاءه ونرغب بإستخدامه يجب تسجيله في ملف الـAndroidManifest.xml الخاص بالمشروع الخاص بنا ..

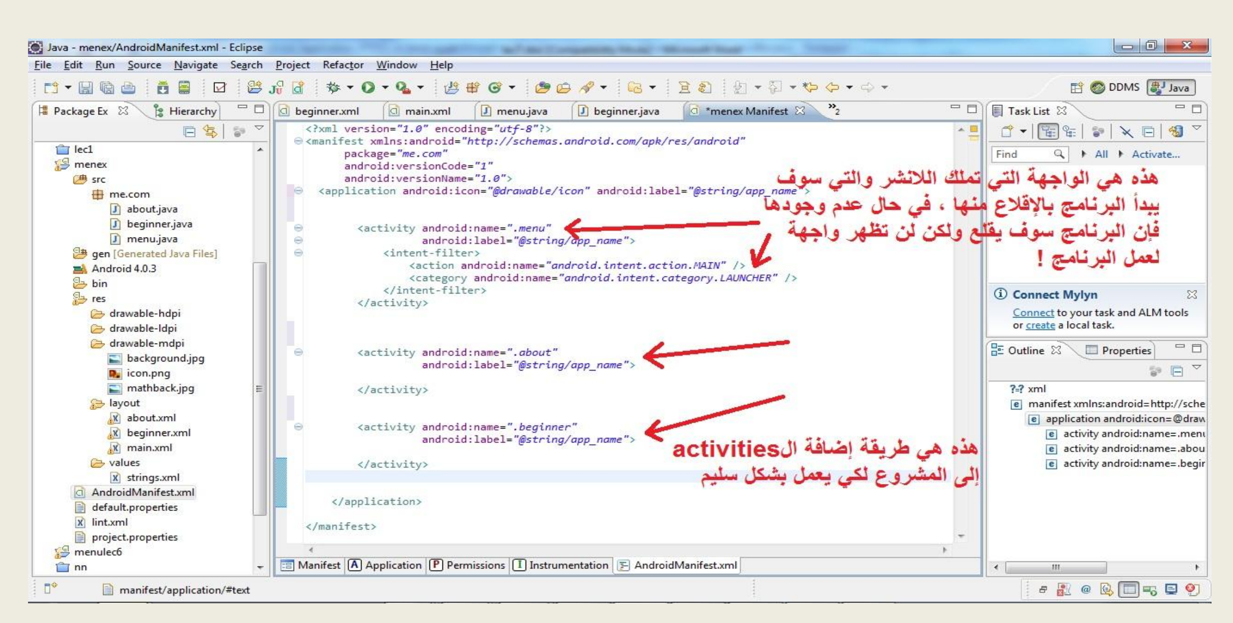
يجب التصريح عن وجود الواجهة وذكر إسمها عن طريق التعليمة :

```
<activity android:name=".test"
                  android:label="@string/app_name">

        </activity>
```

حيث test هو إسم الملف البرمجي والlabel هو الإسم الذي سوف يظهر على إطار الواجهة ..

يجب التصريح عن جميع الواجهات التي نريد إستخدامها كما يجب التصريح عن الواجهة الأساسية التي سوف يبدأ البرنامج بالإقلاع منها ، وهي الواجهة التي تملك الLauncher .. وذلك كما في الشكل التالي :



وأخيرا يصبح المشروع جاهزا للعمل ..

ملاحظات :

1- قد تجد بعض الأكواد البرمجية الخاصة بك خاطئة أو لا تعمل بشكل سليم .. لا تيأس ولا تعجز ، هناك دوما ثغرات بسيطة لا ننتبه إليها في الكود تتسب في ذلك ، دقق وأعد التفكير في الخوارزمية المناسبة حتى وإن إستغرق ذلك وقتا منك .. المهم النتيجة

2- في حال رغبتك بإضافة شيء ما أو بعمل شيء ما بمشروعك ولكن لا تعلم كيف .. لا تقل غير ممكن ، كل شيء ممكن في عالم البرمجة ، كل ما يلزمك هو البحث في الإنترنت بعبارات مفتاحية وسوف تجد ضالتك بكل سهولة ..

أنا شخصيا وانا أكتب هذه المحاضرة قمت بالبحث عن عدة أمور مثل طريقة إضافة كود إلى الEditText ومثل نافذة التنبيه وغيرها .. جميعها أمور بسيطة في حال عجزت قم بالبحث بالإنترنت ..

أشكر جزيل الشكر المساهمين في هذه المحاضرات عن طريق حل مشاكل الأعضاء في موضوع الإستفسارات الخاصة بالدورة

وبالأخص الأخ أحمدlo-as-ve جزاه الله كل خير ..

الوظائف الخاصة بهذه المحاضرة هي كالتالي :

1. إكمال المشروع الذي بدأناه في هذه المحاضرة "لعبة تعليم الحساب بالمستويات الثلاثة ولكن بإضافة بسيطة وهي

في كل مرة نقوم بالدخول على فئة ما "بسيط او متوسط او محترف" يتم توليد أسئلة عشوائية ليتم حلها ، وليس نفس الأسئلة دائما ،بحيث كل فئة يكون بها 10 أو 15 سؤال على سبيل المثال ويتم توليد 4 أسئلة في كل مرة ، أيضا في كل مرة يتم حل جميع الأجوبة وعند ظهور نافذة التنويه والضغط على الزر يتم إعادة وضع أسئلة عشوائية ..

سوف نقوم – إن شاء الله - بتعلم في المرات القادمة مفهوم قواعد البيانات من أجل وضع الأسئلة والأجوبة وتوسيع البرنامج أكثر بشكل إحترافي تحت قاعدة البيانات

2. إكمال المشروع السابق المأخوذ في المحاضرات السابقة :مشروع الأذكار وذلك عن طريق ربط الواجهات التي قمنا بتصميمها عن طريق ال intents ..

سوف نقوم أيضا بالتوسع أكثر في مفاهيم الintent بحيث نقوم لتصميم برامج تستخدم خدمات معيّنة "مثل إستدعاء المتصفح والميديا بلاير وبرنامج الرسائل والإتصال وغيرها ..

بالتوفيق والسلام عليكم ورحمة الله وبركاته

المهندس : عبدالعزيز رضوان

02-may-2012